

Allaire Technical Brief

Secure Web Application Development with ColdFusion 4.0

November 30, 1998

<allaire>

Executive Summary

| | |
|-------------------------|--|
| Title: | Secure Web Application Development with ColdFusion 4.0 |
| Date: | November 20, 1998 |
| Products: | ColdFusion Application Server 4.0 Professional ColdFusion Application Server 4.0 Enterprise ColdFusion Studio 4.0 |
| Target Audience: | Technical Decision Makers IS and IT Managers Web Application Developers |
| Abstract: | ColdFusion is a proven, highly secure environment for web application development and deployment. As with any application development system, a thorough understanding of the security risks and how to address them is essential to ensure a secure application. This document explains the risks associated with ColdFusion application development (and web application development in general), and how ColdFusion developers can address these risks. |

Table of Contents

| | |
|---|-----------|
| EXECUTIVE SUMMARY | 2 |
| TABLE OF CONTENTS | 3 |
| WEB APPLICATION SECURITY CONSIDERATIONS | 4 |
| Security Exposures | 4 |
| Security Solutions | 5 |
| COLDFUSION BASIC SECURITY | 9 |
| Overview | 9 |
| Development | 9 |
| Application Run-time | 11 |
| Server Administration Security | 13 |
| COLDFUSION ADVANCED SECURITY | 14 |
| Overview | 14 |
| Development | 14 |
| Application Run-time Security | 17 |
| Server Sandbox Security | 21 |
| Server Administration Security | 21 |
| CONCLUSIONS | 22 |

Web Application Security Considerations

Web application developers are very concerned about security, and rightfully so. The nature of the web — global access, ease of connectivity and interaction, and lack of true client control — creates an environment where application misuse or abuse can flourish.

As such, almost any discussion of web applications and data integration quickly becomes a discussion of security. Web application developers must fully understand the security risks in order to address the legitimate concerns, while ignoring the tabloid-style hype.

Security Exposures

Web application security risks fall into three major categories:

- Snooping and Eavesdropping
- User impersonation
- Unauthorized access

Snooping or Eavesdropping

The risk of having someone “overhear” data being sent over the Web is a primary concern when sending confidential data, such as credit-card information, over public connections.

On routed IP networks like the Internet, it is relatively difficult to eavesdrop on specific connections without having privileged access to the local ISP routers. Communications packets of any given message could be sent over completely different routes to get from the sender to the receiver, rendering the coherent snooping of the transmission nearly impossible.

The risk of “packet sniffing” is still there, however, especially in Local Area Network (LAN) settings.

User Impersonation

Without proper authentication control, the risk of non-trusted users gaining access to secure information by impersonating trusted users is a very real risk. User authentication

is the foundation of Internet based application security, and inadequate authentication leaves applications vulnerable to attack.

Unauthorized Access

The risk of exposing sensitive information to unauthorized users is the biggest and most complex security risk, because the Internet effectively links every computer to one large network. While completely allowing or disallowing access to a given system or data source remains relatively straight-forward, partial access remains risky. It is this partial access that is frequently required.

For example, it is easy for a large bank like Wells Fargo to put up a public, freely accessible site where NO individual account information is available, but much harder to create an account maintenance site where users have exclusive access to their personal accounts.

Security Solutions

Every Internet based application needs to address these basic security concerns. Over the past few years, several technologies and techniques have evolved as standard mechanisms with which to secure web applications.

ColdFusion integrates with these technologies to provide a seamless, unobtrusive security framework. The risk categories identified earlier:

- Snooping and Eavesdropping
- User Impersonation
- Unauthorized Access

are handled by the following security mechanisms respectively:

- Data Encryption
- User Authentication
- Access Control

An understanding of the basic standard solutions will later clarify the means by which ColdFusion integrates with them to provide a complete security framework.

Data Encryption

HTTP data is often transmitted over open lines, shared data channels, and public access providers. If electronic snoops were to “eavesdrop” on this connection they would be able to copy every byte of data transmitted. While not a common occurrence, this kind of theft is technically possible, making this a legitimate concern, particularly for sensitive data such as credit-card information.

Data encryption is a mechanism used to prevent data from being stolen in-transit between the client and server machines. Electronic thieves can still grab the data being transmitted, but they will find it useless in its encrypted state.

The most common form of data encryption used by web based applications is Secure Sockets Layer (SSL). SSL is a security protocol that provides Internet application protocols (like HTTP) with data encryption using public key cryptography.

Most web servers support SSL, allowing administrators to install a private key that is used to decrypt inbound data and encrypt outbound data. Once installed, the web server automatically encrypts or decrypts data as it is received or transmitted.

Web server connections encrypted using SSL automatically encrypt all communications, including ColdFusion transmissions, without requiring explicit activation of encryption from within ColdFusion.

User Authentication

User authentication is the process of making sure the user is who he claims to be. User authentication is used by all network operating systems (NOS), e-mail packages, and even devices such as ATM machines. Performing user authentication involves prompting the user for a unique identification along with some form of verification – information that no one other than the user could know. Passwords and PIN numbers are common verification mechanisms. This form of authentication is often referred to as “challenge and response.” The application challenges the user to prove his or her identity, and allows access if a valid response is returned.

Some of the more popular challenge response methods include:

- Password Authentication
- Digital Certificates

- Electronic Security Cards

Password authentication is by far the most common, but digital certificates and electronic security cards are rising in popularity.

Password Authentication

In password authentication, the application prompts the user for a user ID and password, which is verified against an account database, and access is either granted or denied based on that verification. While this is the most popular form of authentication, it is also the least secure. The biggest hole in this form of security is the password itself. Recent statistics indicate that most networks do not require users to change passwords frequently, do not prevent password reuse, and do not enforce an adequate minimum password length. Most of the recent high visibility “hacks” into corporate or government networks resulted from poor password protection, allowing electronic thieves the ability to guess passwords and gain access.

Additional problems with password-based authentication include forgotten passwords, and unattended, authenticated workstations. In the first case, the authorized user cannot be unauthenticated while in the latter, the authorized user authenticates himself and then leaves his authenticated workstation available to be accessed by unauthorized users.

Digital Certificates

To solve the problems with passwords, many sites are now taking advantage of “digital client authentication”. These certificates are obtained from a trusted certificate authority, and installed into the browser. The certificate contains signature and key information that can be used to validate its authenticity. This form of authentication provides web servers the ability to query the client browser for a digital certificate, which uniquely and safely identifies a specific browser. Users need not remember passwords; in fact, there are no passwords to remember. The web server and client exchange certificates, and access is granted or denied accordingly. The biggest problem with digital certificates is that they identify clients, not users. They are therefore best suited for users who do not use multiple computers on a regular basis.

Electronic Security Cards

Another popular technology is the use of electronic smart cards or security tokens. There are several variants of these technologies on the market, but they all basically act as

electronic identification cards. These cards require special hardware or software to be installed on the network, and all authorized users are given a card that uniquely identifies them. Some cards are swiped through magnetic readers (similar to credit cards or ATM cards); others display digital IDs that are typed into a computer. These cards can be highly secure; unless lost or stolen, in which case they could be misused.

The common denominator here is that once a user correctly responds to a provided challenge, authentication is complete, and it is safe to assume that the user is who he or she claims to be.

Access Control

Application security is rarely implemented as simply as “deny all access” or “grant all access.” Users are usually granted access to particular features or components based on security clearance, group affiliation, or some other criteria. The process of determining which features and options are visible to specific users is known as “access control.”

Access control typically is used to accomplish one or more of the following:

- Restricting access to an entire application
- Restricting access to specific functions within an application
- Restricting access to specific features within application pages
- Dynamically changing available options to make them user specific
- Restricting access to specific data within an application
- Enforcing levels of access for specific data: no access, read-only, read-write, etc.

As previously explained, web server-based user authentication is used to identify users, and to grant or deny access to entire applications. While this may provide basic security control, it does not provide developers with application level access control. However, because web server user authentication is an ideal method of identifying users, application level access control is often built on top of this.

Unlike data encryption and user authentication, access control is very application-specific. As such, application developers must implement this level of security themselves, inside the application pages.

ColdFusion Basic Security

Overview

The ColdFusion web application development system addresses each of the three aforementioned security concerns at every stage of development, deployment and maintenance.

Two mutually exclusive security models are supported in release 4.0: basic and advanced. Should both security models be activated, advanced security will be used. Should both models be deactivated, all server administration functions and directories become available to all users that have network access to the server. Basic security is activated during the ColdFusion Server installation process by default. The two models are completely independent of one another, and shall be addressed independently.

The basic security in ColdFusion 4.0 provides for the security of all phases of development and deployment, offering a single access level — complete control — for trusted users. Granular run-time access security is both possible and flexible, but involves custom development.

Under basic security, there are three general areas to be secured: development, application run-time, and server administration. For each of these areas, we must consider the three major security concerns: encryption, authentication and access control.

Development Using Basic Security

Overview

The ColdFusion development system introduced the Remote Development Services (RDS) as part of the 3.1 release. RDS was created to allow developers using ColdFusion Studio to develop, deploy, and maintain remote ColdFusion applications on remote ColdFusion servers. When using basic security under ColdFusion 4.0, the development security options look much like they did under 3.1.

| Methods for Accessing Files and Data sources | | |
|--|------------------------------|----------------|
| Method | Description | Security Model |
| LAN based | Uses the windows file system | Encryption |

| Methods for Accessing Files and Data sources | | |
|--|--|---|
| Method | Description | Security Model |
| (file access only) | to provide access to local and network drives. This is the traditional model supported by all web development tools. | <p>Provided by NOS.</p> <p>Authentication</p> <p>Network permissions of user logged into workstation where Studio is being run.</p> <p>Access</p> <p>Controlled by NOS based on permissions granted to user.</p> |
| FTP based (file access only) | Connects to an FTP server running on the same machine as the target ColdFusion server. | <p>Encryption</p> <p>Not available.</p> <p>Authentication</p> <p>Provided by FTP server.</p> <p>Access</p> <p>Permissions defined using the native security of the FTP server software.</p> |
| RDS based (files and data sources) | ColdFusion Studio interacts with the remote file system by connecting to the RDS on the target ColdFusion server. | <p>Encryption</p> <p>Supports SSL.</p> <p>Authentication</p> <p>Controlled by ColdFusion Server. A single password secures the entire server under RDS access, allowing full access to authenticated users.</p> <p>Access</p> <p>All files and mapped network drives on the target server are accessible with single password.</p> |

Encryption

During development, SSL encryption optionally protects all Remote Development System (RDS) communications between the ColdFusion Studio and ColdFusion Server. This includes remote access to server data sources and drives, provided that both are accessed via RDS.

In addition to RDS, application files on remote servers may be accessed via direct file shares or FTP. Encryption for such non-RDS connections is not provided with ColdFusion Studio.

Authentication

RDS and server administration are each secured via a single, independent password. Once successfully authenticated under one of these passwords, users have complete access to all server services supported. Under RDS this involves complete access to all files and data sources available to the ColdFusion server RDS service.

Access Control

Access control at the RDS level involves complete access to the server resources or no access at all. An RDS developer with knowledge of the single RDS password for a given ColdFusion server is granted access to all directories and data sources to which the RDS service on the server has access to.

Application Run-time Using Basic Security

Encryption

All ColdFusion communications with the client browser happen through the web server during application run-time, inheriting all encryption mechanisms that might be active. The addition of encryption to an unencrypted ColdFusion application involves simply adding encryption to the underlying web server.

Authentication

In an environment where multiple developers perform development against the same ColdFusion server, they would all have to share the same development password. In the

case of multiple server administrators they would also have to share a single password. In both cases, no user-specific, graduate access is available.

The authentication of the end-users of ColdFusion applications under basic security is handled through customized user directories created by the developers. External user directories could also be integrated via the various custom extension mechanisms supported by ColdFusion, including CFX tags, and COM or CORBA objects.

Access Control

User access privileges within ColdFusion applications are custom built by the application developer, much like user authentication. They could be based upon a user information tables, or linked to external access control mechanisms through custom extensions.

Basic security leaves most access control mechanisms to be hand-crafted by individual developers via the creation and maintenance of proprietary user tables. Standard web server authentication validates users as discussed above. Once a user is authenticated and granted access, the web server passes the validated user ID to ColdFusion. This allows developers to perform conditional processing driven by the user login.

This conditional processing would typically be performed at an application level, using a special ColdFusion file called application framework page (Application.cfm). ColdFusion automatically processes this file prior to each and every request, making it an ideal location for user authentication.

A typical process flow would be as follows:

- The web server authenticates user.
- The web server passes the user request to ColdFusion along with the name of the authenticated user.
- ColdFusion processes the application page performing a database lookup for name of the already authenticated user.
- “Session” variables are set based on values returned by the database lookup.
- The rest of the application uses these session variables to allow or deny access to specific features.

For example, the application page might perform a database lookup or an LDAP query for the user login to determine if a user has administrative rights. Based on the returned data, a global variable called “IsAdmin” would be set to either Yes or No. The

administrative functions throughout the entire application could now be secured by simply surrounding them by an if statement that checks to see if IsAdmin is Yes.

Server Administration Security Using Basic Security

A single administrative password governs access to the security maintenance features of ColdFusion server under basic security. This single password provides access to the complete ColdFusion administrator, both locally and remotely. The server administration password is separate and distinct from the RDS password.

ColdFusion server administration is web-based, allowing it to inherit the level of encryption offered by its hosting web server. Adding encryption to the web server thus encrypts remote server administration.

ColdFusion Advanced Security

Overview

In addition to supporting the basic security model, ColdFusion 4.0 introduces an entirely new advanced security framework, providing robust, granular security. The new security framework integrates seamlessly with any group of LDAP and NT domain user directories for user authentication, and folds access control security into the server engine. Integration with standard web-server encryption mechanisms provides complete transmission security.

Development Using Advanced Security

Overview

The current ColdFusion 4.0 RDS security model has significantly matured since RDS was first released with ColdFusion 3.1. ColdFusion Server 4.0 introduces an advanced security framework that supports integration with standardized authentication mechanisms and a granular assignment of file and data source access rights.

While security in release 4.0 is robust and vastly improved over release 3.1, some considerations remain:

1. Advanced security requires more configuration and server planning.
2. Basic security is still supported as an alternative to advanced security. The simplicity and lower server security overhead of this alternative come at the price of granular access control and integrated authentication.

These considerations illustrate that no single security model is optimal for every application or development environment. In planning applications, ColdFusion developers must weigh the costs and benefits of the various security alternatives in the context of the project requirements.

It is also important to understand that RDS is an optional feature for connecting to remote servers and that using ColdFusion Studio does not require the use of these services over a network. If customers are uncomfortable with the security implications of RDS on their

server or do not need it, they can secure it with an undistributed password and choose from several other files and database access methods described below.

| Methods for Accessing Files and Data sources | | |
|---|---|--|
| Method | Description | Security Model |
| LAN based (file access only) | Uses the windows file system to provide access to local and network drives. This is the traditional model supported by all web development tools. | <p>Encryption Provided by NOS.</p> <p>Authentication Network permissions of user logged into workstation where Studio is being run.</p> <p>Access Controlled by NOS based on permissions granted to user.</p> |
| FTP based (file access only) | Connects to an FTP server running on the same machine as the target web server. | <p>Encryption Not available.</p> <p>Authentication Provided by FTP server.</p> <p>Access Permissions defined using the native security of the FTP server software.</p> |
| RDS based (files and data sources) | ColdFusion Studio interacts with the remote file system by connecting to the RDS on the target web server. | <p>Encryption Supports SSL.</p> <p>Authentication Transparent LDAP and NT domain user directory integration.</p> <p>Access Granular access control of files and directories to individual groups and users.</p> |

Encryption

Not all development environments have need of encryption. Environments that are wholly shielded by a corporate firewall may have little need for encrypting development communications.

For those that do, ColdFusion Studio offers complete RDS encryption of both data sources and server file access. The connection between Studio and ColdFusion Server can be encrypted using Secure Sockets Layer (SSL), protecting both the data and the login information. Protocols like File Transfer Protocol (FTP) do not embrace encryption, and hardware or performance considerations often make wholesale development encryption unattractive.

Authentication

The advanced security options in ColdFusion 4.0 allow user authentication against LDAP and NT domain user directories. This allows user account maintenance to be centralized, avoiding the maintenance of redundant, application-specific user directories. Netscape Directory Server is bundled with ColdFusion to provide an LDAP user directory for users that don't already have NT domains or LDAP directories in production.

For example, in the case where an entire corporation uses an NT Domain for user network authentication, that same domain can be used to authorize users for access to web applications. The user's network login will be identical to that used for to login to internal web applications.

With the ability to assign authentication privileges at the user group level, web applications based on NT or LDAP user groups *automatically* inherit changes to group membership without requiring any additional maintenance.

The implications of this in the context of the previous example might include the addition of a new HR employee to the NT domain. By simply creating the new user in NT and assigning it to the HRUsers group under the NT domain, all web applications that allow authentication by HRUsers will *automatically* grant him access by virtue of his NT HRUsers group affiliation.

The centralization of user accounts into a very limited number of standard authentication sources simplifies user administration while concentrating the security risk involved with

the exposure of a user login. Rather than an exposed login granting an unauthorized intruder access to a single system, it could potentially grant access to all internal systems to which the compromised user has access privileges.

The use of centralized authentication thus underlines the importance of vigilant login management. More frequent mandatory password changes or the adoption of a more stringent authentication mechanism like electronic security cards could help to address this issue.

Access Control

In addition to using LDAP and NT domain user directories for authentication, ColdFusion 4.0 provides for the assignment of access control to the individual users and user groups from the LDAP and NT user directories. This allows changes to group affiliation and login changes to transparently cascade down to the access privileges of web developers.

Data sources and directories must be initially protected, after which explicit access can be granted to specific groups or individual users. Protected resources remain inaccessible to those users without access to them. Further granularity is provided by the ability to explicitly specify for each data source the types of SQL commands that a given user group may perform against it. For example, on a HR Data source, guests may be allowed to search/select data, while inserts, deletes and updates may be allowed only for HR departmental staff.

Read and write permissions for files and directories on ColdFusion servers can be explicitly assigned at both user and user group levels. You could, for example, grant write access to a development team for their development directory tree alone, and read-only access to them everywhere else.

Application Run-time Security Using Advanced Security

Encryption

As with any data sent over public data lines, information sent to and from ColdFusion applications from clients' Web browsers can be "listened to" unless the communication is encrypted. This is of particular concern when confidential information (such as credit-card numbers) is transmitted. While ColdFusion does not provide connection encryption

itself, it rides transparently upon standard web server encryption. Encryption technologies such as Secured Sockets Layer (SSL) and Virtual Private Networking (VPN) have matured to be extremely secure, and incur minimal performance overhead.

Encryption is not always available desirable due to other considerations, including performance, server and client configuration. Static information pages or non-sensitive material are examples of pages that don't require encryption. Only the specific pages containing sensitive information need to be encrypted.

Most browsers display a special symbol when a page is being sent and received securely (a lock or key is common). The presence of this symbol is important to users, many of whom will not use the site without it.

Authentication

ColdFusion does not have a built-in account management system of its own by design. Instead it is designed to integrate with your web server's own authentication system, NT domain security, and LDAP security.

Web server security is usually either directory or file based. Securing a directory at the web server level also secures all the contents of that directory (and any subdirectories), including any ColdFusion code. That code will be executed only when the web server has successfully authenticated the user and granted him directory access.

ColdFusion 4.0 includes embedded user authentication support for standard NT Domain and LDAP user directories. This allows administrators and developers to more easily integrate authentication into their web applications and across the enterprise without having to create custom modules.

In the case that neither LDAP nor NT Domain security is employed as the authentication standard, ColdFusion continues to support integration with a wide variety of user authentication mechanisms by virtue of its extensibility and flexibility. Integration with existing, non-LDAP and non-NT authentication structures can be achieved through the encapsulation of authentication functionality in custom tags, CORBA or COM objects.

Access Control

Application Security

ColdFusion 4.0 launches a completely new level of access control with its ability to granularly enforce access rights to users and groups from within the server engine. Security is now intimately embedded in the core server functionality.

Users and groups from existing NT Domains, or LDAP user directories can be expressly granted or forbidden run-time access to ColdFusion Applications, CFML tags, collections, components, Data sources, Files, Directories, and Custom Tags.

For example, sensitive tags like <CFREGISTRY> can be restricted for use only by members of the NT Domain Administrators group of the local domain. A sensitive HR search collection could be made available to the HR staff alone, regardless of which ColdFusion application happened to include it. CORBA or COM objects that work with company financial information can be made available only to the departments and web applications that required them. All this can now be enforced by the ColdFusion engine itself. By assigning access rights to NT and LDAP users and groups, web application access security becomes integrated with standard authentication systems.

As the members of those groups change, no additional maintenance is required to add or remove the corresponding rights in the Web Server. Take, for example, a company running on a Windows NT Domain. When a new employee joins HR, that employee would be added to the HR group in the company's NT domain, and would immediately have access to all web application functionality that was already granted to the members of the HR department.

Database Security

Database security is used to control user database access. Database security can happen at two levels: ColdFusion server data source SQL restrictions, and traditional database authentication.

The first level, ColdFusion Server Data source SQL restrictions limits the types of SQL commands that the ColdFusion engine will allow to be sent to the Data source. If the current web application user is only permitted to enter SQL SELECT commands against the HR database, for example, all of his attempts to perform salary UPDATES will be gated right within ColdFusion, before they are even sent off to the HR Data source.

This is a revolutionary step ahead of the earlier ColdFusion security model. Never before could such permissions be directly associated with LDAP and NT Domain user groups directly from the server. Instead, such rights were hard-coded into the Data source definition.

The second level, database authentication involves sending a username and password to data sources during run-time as part of the authentication parameters. The result of the request will be contingent on the rights assigned to that user login within the database. To take our HR example a little further, the salary column of a SQL database table might be secured for viewing by the entire HR department, but accept updates from the HR director alone. In the case that somebody in HR other than the director attempts to perform a salary update, the request would be sent to the SQL database along with the user login information. The SQL server would authenticate the user to learn that he didn't have the necessary access rights, and would deny the update request.

In addition to standard ODBC Data sources, ColdFusion now supports native drivers for Oracle and Sybase, and OLE-DB for Microsoft SQL Server. The same data security technology works across data source types.

For customers that choose not to use ColdFusion's new advanced security features, the traditional mechanisms of setting user login rights in the Data source definition are still valid. However, to further secure their databases, ColdFusion administrators can take several precautions:

- Data sources should typically not be defined with administrative level access to their underlying databases.
- Access should be granted to only those databases, views, stored procedures, and operations absolutely required by the application.
- Many developers define multiple data sources to the same database, each with different user logins and thus different levels of data access. Applications that need greater levels of data access would use an ODBC data source with higher login rights, and those applications would be further secured using the user authentication and access control techniques mentioned earlier.

It is important to realize that your data is only as secure as the level of access granted to users. Freely allowing all users to write any SQL code for processing by ColdFusion could seriously compromise database security.

Server Sandbox Security

ColdFusion 4.0 introduces the concept of security sandboxes. A security sandbox assigns security permissions to a directory tree. These rights override any other security rights individuals might have for that specific directory tree.

For example, a given directory `/webapps/hr_view_app` might be designated as a sandbox in which database write permissions to HR data sources are denied. All users of applications physically located in that directory would be denied write permissions to HR data sources, regardless of their personal access privileges. Even the HR director, who may have write permissions to HR data sources in all other contexts, would be constrained from using them in this restricted sandbox.

This is particularly useful in Web Hosting situations or in circumstances where disparate functional groups share an Application Server. Server administrators can deploy multiple applications on the same server without creating the risk that one application will access another application's resources.

For example, an ISP hosting two different domains hosted on a single box may well want to restrict the custom tags and datasources available to each of the domains. This could allow each domain to submit their own custom tags and create their own data sources to the to which they are given exclusive access. The same is true for all other securable object types and ColdFusion functionality.

Server Administration Security

Server administration under ColdFusion 4.0 is designed to permit the decentralization of server management. Each ColdFusion server supports multiple distinct managers with various rolls. Authentication of the administrators is performed against the LDAP or NT Domain user directories with three levels of administration privileges: CFAdministrators, CFPrivileged, and CFRestricted.

Administrators have complete access to the administrator, including the ability to administrate security. Privileged users have access to everything in the ColdFusion administrator page *except* security items. Restricted users can see no more than select data sources and verity collections in the ColdFusion administrator.

Conclusions

As the web ushers in a new era of computing that mixes public, private, and semi-private networks supporting a broad range of business and e-commerce systems, security is of the utmost concern in the web environment.

ColdFusion provides a complete web application development system with total system security from development through deployment. It provides support for a variety of encryption, authentication, and access control technologies at each stage of the production cycle.

The 4.0 release adds advanced security with standard LDAP and NT Domain authentication to the realm of the core server, allowing for the granular assignment of access security based on those standard logins.

The basic security model in release 4.0 will continue to be actively used by ColdFusion developers old and new alike. Although it doesn't offer the same granularity of access control or integration with external user authentication mechanisms, it provides for user defined security mechanisms that may be optimal for certain applications.

For corporate web applications where most users are internal and authenticated, and NT Domains or LDAP user directories are already in use, the advanced security model offers more granular and seamless integration to existing mechanisms. For those applications that are completely independent from existing corporate security infrastructure and turn-key solutions, basic security may be more appropriate.

ColdFusion continues to offer a standards-based open architecture. Release 4.0 integrates a robust, engine-level security framework while continuing to support customer-specific security solutions.